

Homework 5

Solution Key

Problem 5.4

In the course directory is a file named `hwk5_mystery.c`. ... Your goal is to determine precisely how the C compiler and Mac OS cause the memory to be laid out at the point marked `/ HERE */`. ...*

My code and the corresponding output are in the course directory, under `sol/hwk5/`, but the output is also presented on the next page, somewhat formatted. Specifically, the bytes are in a table with four bytes per line. Preceding the line is an 8-digit hex string that gives the 32-bit pointer to the first byte in the line. For each byte, a decimal representation is given, and this is followed by a representation of the corresponding character if the byte represents a char, or by the corresponding two-digit hex number if the byte represents a portion of a pointer. Finally, if the bytes on the line have a clear interpretation with respect to the program, this is given at the end of the line.

The ones marked with an asterisk are not values in current use, and at least some are presumably remnants from the call to `printf`. Quite a few bytes aren't marked at all, because this isn't CS 201 and we don't have to decipher every detail. The key pieces to look at are the layout of the array of snorks and of the individual snorks in that array, and the way that the actual declared variables and parameters are laid out.

| | | | | | |
|------------|---------|---------|---------|---------|--|
| 0xbfff5f4: | 97 'a' | 98 'b' | 99 'c' | 100 'd' | char test[8] |
| 0xbfff5f8: | 101 'e' | 102 'f' | 103 'g' | 0 '\0' | |
| 0xbfff5fc: | 252 fc | 245 f5 | 255 ff | 191 bf | char *ch |
| 0xbfff600: | 0 | 0 | 0 | 0 | |
| 0xbfff604: | 47 | 31 | 0 | 0 | struct snork *snp |
| 0xbfff608: | 104 68 | 246 f6 | 255 ff | 191 bf | |
| 0xbfff60c: | 186 | 31 | 0 | 0 | |
| 0xbfff610: | 56 38 | 246 f6 | 255 ff | 191 bf | |
| 0xbfff614: | 0 | 0 | 0 | 0 | |
| 0xbfff618: | 0 | 0 | 0 | 0 | |
| 0xbfff61c: | 0 | 0 | 0 | 0 | |
| 0xbfff620: | 0 | 0 | 0 | 0 | |
| 0xbfff624: | 0 | 0 | 0 | 0 | |
| 0xbfff628: | 7 | 0 | 0 | 0 | |
| 0xbfff62c: | 104 'h' | 105 'i' | 106 'j' | 107 'k' | start of a[0].second |
| 0xbfff630: | 0 '\0' | 0 | 8 | 0 | last byte of second, pad byte, a[0].third |
| 0xbfff634: | 9 | 0 | 224 | 143 | a[0].fourth, two pad bytes |
| 0xbfff638: | 10 | 0 | 0 | 0 | a[1].first |
| 0xbfff63c: | 108 'l' | 109 'm' | 110 'n' | 111 'o' | start of a[1].second |
| 0xbfff640: | 0 '\0' | 0 | 11 | 0 | last byte of a[1].second, pad byte, a[1].third |
| 0xbfff644: | 12 | 0 | 0 | 0 | a[1].fourth, two pad bytes |
| 0xbfff648: | 13 | 0 | 0 | 0 | a[2].first |
| 0xbfff64c: | 112 'p' | 113 'q' | 114 'r' | 115 's' | start of a[2].second |
| 0xbfff650: | 0 '\0' | 0 | 14 | 0 | last byte of a[2].second, pad byte, a[2].third |
| 0xbfff654: | 15 | 0 | 0 | 0 | a[2].fourth, two pad bytes |
| 0xbfff658: | 192 | 4 | 5 | 0 | two pad bytes?, meef's short n |
| 0xbfff65c: | 6 | 0 | 0 | 0 | int p |
| 0xbfff660: | 4 | 0 | 0 | 0 | addr of low-order byte of n, below * |
| 0xbfff664: | 76 4c | 247 f7 | 255 ff | 191 bf | |
| 0xbfff668: | 152 78 | 246 f6 | 255 ff | 191 bf | int m |
| 0xbfff66c: | 217 | 31 | 0 | 0 | |
| 0xbfff670: | 3 | 0 | 0 | 0 | addr of main's int n * |
| 0xbfff674: | 140 8c | 246 f6 | 255 ff | 191 bf | |
| 0xbfff678: | 140 | 0 | 0 | 0 | low-order byte of main's int n * |
| 0xbfff67c: | 0 | 0 | 0 | 0 | int n |
| 0xbfff680: | 0 | 0 | 0 | 0 | |
| 0xbfff684: | 0 | 0 | 0 | 0 | |
| 0xbfff688: | 75 | 21 | 224 | 143 | |
| 0xbfff68c: | 4 | 0 | 0 | 0 | |
| 0xbfff690: | 0 | 0 | 0 | 0 | |